

Toward Generic, Immersive, and Collaborative Solutions to the Data Interoperability Problem which Target End-Users

Arturo Sánchez-Ruíz and Karthikeyan Umapathy

School of Computing, University of North Florida

1 UNF Drive, Jacksonville, FL 32224, USA

{asanchez, k.umapathy}@unf.edu

Pat Hayes

Florida Institute for Human and Machine Cognition

40 South Alcaniz Street, Pensacola, FL 32502, USA

phayes@ihmc.us

Received 25 February 2009; Accepted 25 March 2009

In this paper, we describe our vision of a “just-in-time” initiative to solve the Data Interoperability Problem (a.k.a. INTEROP.) We provide an architectural overview of our initiative which draws upon existing technologies to develop an immersive and collaborative approach which aims at empowering data stakeholders (e.g., data producers and data consumers) with integrated tools to interact and collaborate with each other while directly manipulating visual representations of their data in an immersive environment (e.g., implemented via Second Life.) The semantics of these visual representations and the operations associated with the data are supported by ontologies defined using the Common Logic Framework (CL). Data operations gestured by the stakeholders, through their avatars, are translated to a variety of generated resources such as multi-language source code, visualizations, web pages, and web services. The generality of the approach is supported by a plug-in architecture which allows expert users to customize tasks such as data admission, data manipulation in the immersive world, and automatic generation of resources. This approach is designed with a mindset aimed at enabling stakeholders from diverse domains to exchange data and generate new knowledge.

Categories and Subject Descriptors: D.2.12 [**Software Engineering**]: Interoperability–*datamapping, interface definition languages*; D.2.11 [**Software Engineering**]: Software Architectures–*data abstraction, languages (e.g., description, interconnection, definition), patterns (e.g., client/server, pipeline, blackboard)*; H.5.1 [**Information Systems**]: Multimedia Information Systems–*animations, artificial, augmented, and virtual realities*; H.5.3 [**Information Systems**]: Group

Copyright(c)2009 by The Korean Institute of Information Scientists and Engineers (KIISE). Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Permission to post author-prepared versions of the work on author's personal web pages or on the noncommercial servers of their employer is granted without fee provided that the KIISE citation and notice of the copyright are included. Copyrights for components of this work owned by authors other than KIISE must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires an explicit prior permission and/or a fee. Request permission to republish from: JCSE Editorial Office, KIISE. FAX +82 2 521 1352 or email office@kiise.org. The Office must receive a signed hard copy of the Copyright form.

and Organization Interfaces—*computer-supported cooperative work, synchronous interaction, web-based interaction*

General Terms: Design, Human Factors, Languages, Algorithms

Additional Key Words and Phrases: Data Interoperability, INTEROP, End-User Approaches, Immersive Collaboration, Plug-in Architectures, Second Life, Ontologies, CL, IKL

1. INTRODUCTION

The IEEE Standard Computer Dictionary defines interoperability as the “ability of two or more systems or components to exchange information and to use the information that has been exchanged” [IEEE Computer Society 2001]. Instances of this problem abound, and will continue to be the topic of research in the context of a wide variety of critical application domains.

For instance, consider an excerpt from the IEEE-USA position in connection with the National Health Information Network: “interoperability is one of most critical concepts confronting the adoption and implementation of enhanced electronic information technologies into our national healthcare infrastructure” [IEEE-USA 2005]. Consider also what the US Defense Standardization Program (DSP) states in its website: “The need for more interoperability and information sharing across DoD [Department of Defense] in support of Joint Service, and multinational operations continues to be emphasized in the National Military Strategy” [US Department of Defense 2005].

The following fictitious scenario illustrates an instance of the Data Interoperability Problem (a.k.a. INTEROP.) The emergency rooms of various hospitals in county Y of state X (collectively referred to as hospitals producers of data or HPD) collect information related to treated cases. The office of crime prevention of state X’s police department (SXP) recently acquired an expensive simulator which, given the appropriate data, produces geographical maps of state X showing various indicators which forecast crime by county, city, and neighborhood. An instance of INTEROP is: how can SXP use the data produced by HPD to feed its simulator and produce maps which can be used to support decision making within SXP and state X?

In the general INTEROP, data are characterized by syntactic and semantic elements. Data generated by producers are characterized by PD (which stands for the producer’s characterization of said data,) while data required by the consumers are characterized by CD (which stands for the consumer’s characterization of said data.) INTEROP arises when PD and CD do not directly align, that is to say: the way consumers characterize needed data (according to CD) does not directly correspond to the way producers characterize the counterpart available data (according to PD.) In other words, valuable data needed by consumers is available from the producers but it is not evident how to use them.

A success INTEROP story in the domain of data exchange among disparate design and manufacturing systems is that behind the Standard for the Exchange of Product Model Data, or STEP, which is ISO standard 10303 [STEP Tools Inc. 2008]. A standard-based solution to the interoperability problem is but one approach among many that have been proposed in connection with various application domains.

In this paper, we present our vision of an approach to INTEROP we refer to as “just-in-time”, which tells the stakeholders “come as your are”, and empowers them

with a suite of integrated tools that enable interactive, real-time collaboration in an immersive virtual world, à la Second Life [Linden Lab 2008a], via direct manipulation of data representations. Examples of resources that can be automatically generated by the immersive world, from specifications gestured by stakeholders (through their avatars,) include (but are not limited to): multi-language source code, visualizations, web pages, and web services. The generality of this approach stems from the fact that customizations are supported by a plug-in architecture, à la Eclipse [Eclipse Project 2008]. Alternative labels for our approach we interchangeably use are: “real-time”, “on-demand”, “right-here-right-now”, and “agile”.

The rest of the paper has been structured as follows: in section 2 we discuss levels of interoperability and approaches to INTEROP. Section 3 presents an architectural vision of our approach with pointers to existing approaches and technologies that can be either used as an exemplar, or directly utilized, to implement such vision. Our paper concludes with section 4 which reports on the current status of our vision.

2. LEVELS OF INTEROPERABILITY AND APPROACHES TO INTEROP

A fundamental objective of any data interoperability approach is to provide effective access to a wide variety of autonomous and heterogeneous data sources. These approaches have to cater to data sources that may contain structurally and semantically diverse information and may need to support different access methods to interface with them. A data interoperability approach should permit integrated querying and transformation of diverse information while preserving autonomy and integrity of respective data sources [Vassalos 2004]. In this section we start by discussing the concept of levels of interoperability. We then present a brief description of some approaches to data interoperability followed by a discussion on the need for flexibility and agility as desired traits of any approach to interoperability.

2.1 Levels of Interoperability

Tolk and Muguira [Tolk and Muguira 2003] originally proposed a so-called “Levels of Conceptual Interoperability Model” (LCIM) as a 5-tiered reference framework to assess the extent to which data interoperability was supported by military modeling and simulation applications. LCIM has evolved into a 7-tiered model, which is documented elsewhere [Turnitsa 2005] (see Table I). This extended LCIM, which can be used for any domain, also categorizes interoperability levels on three dimensions based on the work by Page et alia [Page et al. 2004]. The generalized LCIM is an attempt to offer insight into better understanding the gap between human conceptualizations of application domains and the actual raw data that is exchanged among systems [Winters et al. 2006].

2.2 Summary of Some Approaches

This section presents brief summaries of some salient approaches to INTEROP.

2.2.1 Database integration The database integration approach provides unified access to data which reside in multiple databases [Lenzerini 2002]. This approach takes the

Table I. Extended Levels of Conceptual Interoperability Model from [Turnitsa 2005].

Extended LCIM	
Levels	Categories
Level 0: No Interoperability (e.g., Stand-alone systems) Level 1: Technical Interoperability (e.g., Communication protocol for exchanging data)	Integration
Level 2: Syntactic Interoperability (e.g., Data format for exchanging information) Level 3: Semantic Interoperability (e.g., Reference model to interpret meaning of data) Level 4: Pragmatic Interoperability (e.g., Data and its context are unambiguously defined)	Interoperation
Level 5: Dynamic Interoperability (e.g., Changes to data and its subsequent effects are unambiguously defined) Level 6: Conceptual Interoperability (e.g., Data in a form independent of implementation models)	Composition

data schemas of each database as input and produces an integrated data schema along with associated mapping of data distributed on multiple databases [Parent and Spaccapietra 2000]. Under this approach users are allowed to post queries based on the integrated data schema which are reformulated in terms of a set of queries over respective databases [Lenzerini 2002]. Database integration is a suitable approach for achieving data interoperability when all data reside in database management systems and not in any other sources such as files and spreadsheets. Database integration poses complex problems such as: the creation of appropriate integrated schema; the assessment of integration rules and their properties; the resolution of fragmentation conflicts; the optimization and evaluation of queries; and the direct integration of heterogeneous databases [Lenzerini 2002; Parent and Spaccapietra 2000].

2.2.2 Standards One of the oldest approaches to solve the interoperability problem is to agree on and then build a compatible standard [Paepcke et al. 1998]. Standards allow us to achieve data interoperability among products and across organizations [West 2003]. Standardization approaches can be classified into two categories, *de jure* and *de facto*. *De jure* standards are legally enforced specifications either by the government or by non-profit standardization organizations such as ANSI [American National Standards Institute 2008]. Most *de jure* standards are developed in an open consensus-based process where various stakeholders come together to decide on crucial elements of the standard to achieve interoperability. In contrast, *de facto* standards are decided based on, for instance, adoption rates and market dominance among other competing standards. Many *de facto* standards are proprietary technology as they are developed by individual organizations. Even though standards are a key enabler of interoperability, developing standards for and across domains is a tedious

and time-consuming iterative/evolutionary process which naturally tends to leave open issues that need to be resolved in future phases [TechNet 2001].

2.2.3 *Lingua-franca* Under the lingua-franca approach, data interoperability is achieved by translating datasets into one common data model that describes all data from multiple data sources. This approach offers advantages such as common representation of global data, support for multiple views of the same content, separation from the core of existing systems, freedom to transfer all data into a single database, ease of integration with future applications, and manageability as data sources/datasets evolve over time [Wilkes 2003]. XML has emerged as a popular *de facto* lingua-franca infrastructure for data interchange as it provides flexibility to define both structured and semi-structured data, as well as the ability to add tags which carry semantic information. Despite of possessing these positive traits, XML still falls short with respect to support for ad hoc queries, transactional updates, and subscriptions to specific kinds of changes, in comparison to relational databases [Seligman and Rosenthal 2001].

2.2.4 *Ontologies* One of the most popular definitions of the term ontology, as a denotation of a design artifact, characterizes it as “an explicit specification of a conceptualization” [Gruber 1993]. For pointers to more recent characterizations of the term, see [Sánchez-Ruíz et al. 2007]. Fundamental elements of an ontology are a set of concepts, their properties, and inter-relationships. Because of this, ontologies have been considered by some authors as repositories of concepts akin to relational databases [Dell’Erba et al. 2002]. Ontologies are useful for achieving data interoperability as they provide a framework and contextual knowledge required for data transfer and translation between systems [Visser et al. 2000]. Many applications developed in diverse domains have shown that well crafted ontologies can help in achieving efficient data interoperability [Doerr 2004]. However, ontologies with partial or incomplete knowledge would cause data inconsistencies and integration conflicts [Amit 2004]. Such situations are sometimes unavoidable when ontologies are designed by individuals from diverse domains with the goal of modeling multiple data sources.

2.2.5 *Advanced metadata* The concept of metadata, generally defined as data about data, describes characteristics of data such as object names, statement about objects, type of data, and relationships among data. It therefore enables the process of interpreting data [Bretherton and Singley 1994]. Advanced metadata contains extended characterization of data which includes details on processes, systems, functionalities, and transformation specifications [Winters et al. 2006]. Thus, advanced metadata provides full semantic context of the data which can be used to achieve data interoperability. This approach is still in its infancy and considerable work needs to be done on organizing, maintaining, and defining metadata with the goal of producing viable implementations [Winters et al. 2006].

2.2.6 *Services as Data Wrappers* Web services are currently one of the most preferred approaches to solve the interoperability problem among software applications

[Umaphy and S. Puroo 2006]. Conventional data interoperability approaches, such as data warehousing and database integration, do not support realtime access to data and are costly to implement [Pan and Viña 2004]. As an alternative, data sources can be wrapped as services and made available on demand [Riedemann and Timm 2003]. For instance, when a user query is received, it can be decomposed into sub-queries over corresponding data sources which can be individually executed and then combined to provide a global answer to the user [Pan and Viña 2004]. While this is a very promising approach, it is currently limited by the lack of fully dynamic discovery and binding processes in the context of large-scale environments [Zhu et al. 2004].

2.2.7 Data warehouses Data warehouses have been described as a “subjectoriented, integrated, time-varying, non-volatile collection[s] of data” primed for supporting decision making [Chaudhuri and Dayal 1997; Inmon 1992]. At the core of a data warehouse one finds a database that stores operational data and which has been optimized for the generation of data views, the processing of data queries, and facilitation of data analyses [Calvanese et al. 2001]. It typically consists of two set of components. The first set includes tools for extracting, transforming, and loading data into the warehouse [Rifaieh and Benharkat 2002]. The second set includes tools for retrieving and analyzing data for decision making [Rifaieh and Benharkat 2002]. In order to successfully utilize data warehouse tools, metadata mapping between all data sources must be implemented. Given that data warehouses are used for dealing with large scale datasets, maintenance of these tools and metadata mapping become hard tasks as data sources/datasets evolve over time.

2.3 The Need for Flexibility and Agility

The moment at which interoperability is actually addressed can be categorized into two kinds: before-the-fact and after-the-fact. Before-the-fact interoperability occurs before the components to be integrated as a cohesive system are actually developed, time at which all involved parties can agree on models and levels of interoperability among such components. This demands from developers to be in close interaction with each other. Typically, before the fact interoperability is approached by means of tight design protocols, thus the resulting integrated system runs the risk of being inflexible to new additions and might not provide end users with the capability of creating ad hoc solutions [Carney et al. 2005].

After-the-fact-interoperability occurs when developers face the problem of integrating independently built components. This scenario can be found in environments where operational contexts are continuously changing, therefore, parties involved do not know a priori how systems will interoperate. Hence agreements on interoperability must be negotiated and reached dynamically. After-the-fact approaches should be agile enough as to provide end users with assistance to help them solve problems that may occur due to changing contexts by, for instance, creating ad hoc integration solutions [Carney et al. 2005].

These two moments have been characterized as “reusability scenarios” in [Sánchez-Ruiz 1995]. The second moment has been characterized as “post facto” integration in [Power 1990].

With respect to the extended LCIM (see Table I), current data interoperability approaches address integration and interoperation dimensions, i.e., technical, syntactic, semantics, and pragmatic aspects of data interoperability. Thus, most of current data interoperability approaches provide capability to achieve before the fact interoperability. To achieve after the fact interoperability, addressing integration and interoperation is not sufficient; composition aspects must also be addressed. Composability can be achieved when both data modelers and domain experts communicate to develop a common conception of reality surrounding data and system [Hofmann 2004; Tolk and Muguira 2003]. However, most data modelers abstract and simplify models based on their perception and conception of reality which might be different from that of domain experts [Hofmann 2004]. Most of the current data interoperability approaches are not suitable for after-the-fact interoperability because they do not support agile communication and interaction among end-users, data modelers, and other domain experts.

In next section, we describe our “just-in-time” approach which is designed to support after the fact interoperability targeting end-users.

3. AN ARCHITECTURAL PERSPECTIVE OF OUR APPROACH

The first part of this section discusses the main architectural components of our approach. The second mentions approaches and technologies which are either considered exemplars, or can be directly used in the implementation of our architectural vision.

3.1 The Architecture and its Components

Figure 1 depicts the main components of an architectural rendering of our just-in-time approach to INTEROP.

Stakeholders own data which are exposed to others via *Data Sources* with a variety of formats, including but not limited to: non-structured files (e.g., plaintext files), structured files (e.g., using XML), relational databases implemented by specific database management systems (e.g., MySQL, MS SQL Server, Oracle, et cetera), and streams (e.g., real-time-generated data).

Stakeholders cooperate in real-time through their avatars in a immersive virtual world (IVW) implemented à la Second Life [Linden Lab 2008a]. Since the IVW must provide the avatars with the ability to directly manipulate the data associated with the INTEROP of interest to stakeholders (both producers and consumers), data must be *admitted* into the IVW.

The *Data Admission* layer offers stakeholders patterns that can be used to map their data sources to elements in the *Metadata Model*, whose semantics are defined by ontologies which conforms to the Common Logic Framework [Common Logic Working Group Website 2008; ISO/IEC 2008]. We use the term *Metadata Model* to signify a Meta-Model for the Data admitted into the IVM. For other definitions which frame our work see [Sánchez-Ruiz et al. 2007]. The *Metadata Model* and its supporting ontologies therefore give semantics to data admission patterns, data manipulation gestures, and supports translations to other representations (e.g., source code written in a variety of programming languages.)

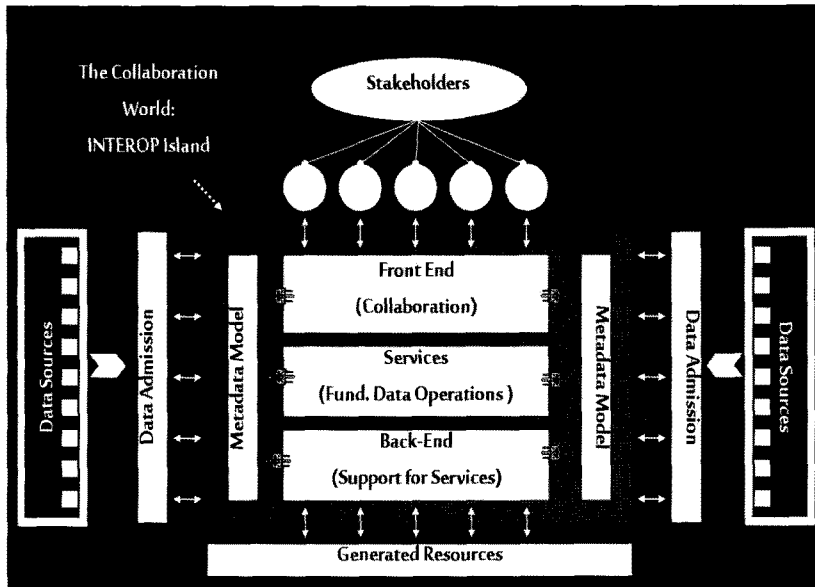


Figure 1. Architecture of Our Approach.

Data types supported by the *Metadata Model* include primitive types with various levels of accuracy (e.g., float and double); multidimensional arrays (including sparse arrays and associative arrays); trees (e.g., to represent taxonomies); general relational networks (e.g., to represent relational data models); and aggregations (e.g., records or structures). Fundamental operations on these types include arithmetic and logical operations defined on primitive types; selection (including slicing and splicing); comparisons (e.g., shallow and deep); cloning; searching; navigation (including traversals); plotting, and general transformations (e.g., matching).

The *Front-End* layer implements all the direct manipulation metaphors which allow users to admit data and collaborate within the IVW. Its core is constituted by visual representations of supported data types and gestures which implement fundamental operations on them which allow users to visually describe resources the IVW is to automatically generate on demand.

For instance, the type 2-D array can be visually represented using the classical rows-and-columns grid. The type 3-D array can be visually represented as a cube. Trees can be represented as hierarchies whose levels can be expanded and contracted. Aggregations can be represented as boxes whose contents can be selectively inspected. General networks can be represented as grids of nodes and roads which can be navigated. Graphical icons would suggest users what can be done with the visual representation in question. For instance, rotating the cube that represents a 3-D array, and splicing a region of a general network. The *Front-End* layer is therefore equivalent to the Graphical User Interface (GUI) of classical web-based, desktop-based, and mobile-based applications.

The *Services* layer provides the actual implementations, as supported by the *Metadata Model*, of all fundamental operations connected to visual metaphors

available to avatars.

The *Back-End* layer interacts with the Services layer to automatically generate all *Resources* supported by the IVW which represent what stakeholders obtain as the result of their cooperation within the collaboration world. These resources include: multi-language programming components (e.g., classes, methods, procedures, and functions written in various programming languages); web services, web “faces” or “skins” (e.g., implemented as facebook applications [facebook Developers 2008]); web applications (e.g., implemented as “mashups” [Some Mashup Resources 2008]); additional data sources (e.g., the product of transforming input data sources); and knowledge (e.g., record of decisions made, transformations, representations, patterns, matchings, etc.)

The generality of this approach is supported by a *Plug-in Architecture*. For instance, to achieve generality across application domains, expert users—in cooperation with domain users—can define the Metadata Model with the fundamental domain concepts and their relationships. Data admission patterns, data representation schemes, and fundamental operations on the data are then implemented as supported by the Metadata Model. Some generative services can be reused across application domains, such as multi-language source code. However, specific plug-ins need to be developed for specific services.

3.2 Related Approaches and Technologies

3.2.1 Immersion The prototypical technology that can be used to implement the immersive, interactive, and collaborative aspects of the IVW is Second Life [Linden Lab 2008a]. Visual metaphors are implemented using the Linden Scripting Language (LSL) [Linden Lab 2008c]. Linden Lab offers services associated with leasing “real state” on which the IVW is built [Linden Lab 2008b]. Examples of collaborative worlds that have been developed in Second Life include the National Oceanic and Atmospheric Administration’s (NOAA) Meteora and Okeanos Islands [Meteora and Okeanos Islands 2008], and NASA’s Space CoLAB [NASA’s Space CoLAB 2008]. They enable avatars to cooperate via textual and aural chatting, and to interact with previously programmed objects. These two projects are associated with SciLands, a community which develops Second Life islands “devoted to science and technology” [SciLands Virtual Continent 2008]. Alternative technologies which can be used to develop the IVW include OpenSimulator (a.k.a. OpenSim) [OpenSimulator Wiki 2008], and Sun Microsystem’s projects Wonderland [Sun Microsystems 2008b], and Darkstar [Sun Microsystems 2008a].

Immersive technologies such as Second Life are a conduit which facilitate the implementation of cooperation models on, for instance, the web. However, they are not a substitute for actual cooperation models. We are currently using [Stahl 2006; Kaptelinin and Czerwinski 2007] as references to design and implement various cooperation models.

3.2.2 Plugins Eclipse has become a well-established, exemplary, and popular Interactive Development Environment (IDE) which was built around the motto: “for anything, but nothing in particular” [Eclipse Project 2008]. Other IDEs have taken this initiative

as a paradigm to follow; for instance Sun Microsystems' Net-Beans [NetBeans Plugin Portal 2008], and Microsoft's Visual Studio (where plugins are called "add-ins") [Skibo et al. 2006]. The very attractive and effective "Lego metaphor" behind its conception has its roots in the work of the OSGi Alliance [OSGi Alliance 2008]. The plugin architectural pattern is of course not a panacea, and we are aware of some recommendations which we are using in the implementation of our approach (see, for instance, [Birsan 2005] and [Kriens 2008]). Two of the most interesting challenges stem from the fact that not all plugins need to be in the same address space, unlike the case of a desktop application, and that access to the IVW might imply the execution of multiple threads, as in the case of massively multiplayer online games (MMOG).

3.2.3 Ontologies Ontologies give semantics to our Metadata Model and we propose using the Common Logic Framework (CL)—which is ISO/IEC-24707 standard as of 2007—to express them. CL offers a rich platform equipped with abstract syntax and semantics which can be used to define languages (called CL dialects) on which formal ontologies are written [ISO/IEC 2008]. The standard itself ships with three dialects: Common Logic Interchange Format (CLIF), Conceptual Graph Interchange Format (CGIF), and XML for Common Logic (XCL). We are actually using IKL, which is an extension of CL, with a syntax similar to that of CLIF, and semantics which support the "talking about propositions as well as simply expressing propositions by sentences." [Hayes 2008a; Hayes 2008b]. We are interested in structuring the ontologies that give semantics to the Metadata Model around pattern languages which help bridge the gap between the intuitive intention of the ontology engineer and the actual formal terms in the ontology [Sánchez-Ruiz and Hansen 2005]. Although description logics such as OWL-DL [Smith et al. 2004] allow for guaranteed inference performance, they suffer from severe expressive restrictions which are semantically arbitrary, and often impede ontology design. As a common example, OWL cannot describe the relationship of being an uncle in terms of those of father and brother, because the definition has an embedded existential quantifier. Many applications of OWL require the implementation of ad-hoc inference machinery (using 'rules' or some such) to overcome such limitations, running a grave risk of introducing semantic errors. Common Logic is as expressive as anyone can want, while still allowing description logics (and other special-case logical restrictions) to be smoothly represented with in the CL syntax as recognizable special cases (using, if required, the same recognizable vocabulary as OWL, for example.) In this way, Common Logic provides a single all-encompassing logical framework with a single coherent semantics. The clarity and internal coherence of the CL system is its chief operational advantage.

3.2.4 Data mapping Two commercial products which target software developers, Altova's Mapforce [Altova 2008], and Microsoft's SQL Server Integration Services [Microsoft 2008], offer their users a wide variety of data mapping and integration functionality presented through a visual (drag-and-drop) interface. Our approach is a generalization of these products in that it targets data stakeholders, not software developers, supported by a plug-in architecture.

3.2.5 Resource generation To generate the code associated with resources, an intermediate representation layer is being defined from which modules written in specific programming languages can be created. Specific translation patterns are implemented by plugins. This approach is based on an earlier work which aimed at the automatic generation of language wrappers to enable multi-language interoperation, i.e., the ability of a module written in language L_1 to use a module written in language L_2 , when both languages are not the same, without resorting to inter-language translation [Sánchez-Ruíz 1995].

3.2.6 Virtual solar-terrestrial observatory The so-called Virtual Solar-Terrestrial Observatory (VSTO) is a very interesting project funded by the National Science Foundation which has goals that overlap with ours, namely: it targets end-users, it is ontology-driven, and it has generative components. However, in its current implementation real-time cooperation among data stakeholders is not supported, and its software architecture does not seem to be extensible. VSTO acts as a much needed point of confluence among scientists from different disciplines by assimilating heterogeneous data through a OWL-based ontology using which workflow-driven semantic queries can be answered to lead end-users to the desired data [McGuinness et al. 2008].

4. CURRENT STATUS

Our first case study is associated with the Florida Mesonet, a network of wireless sensors which capture land and coastal high resolution weather data from various state and federal agencies to be shared among various state and federal stakeholders. This project has been funded by NOAA through the FL Hurricane Alliance [Florida Hurricane Alliance 2008]. Data models which are being studied include the ones from the CLARUS project (US Department of Transportation) [US Department of Transportation 2008], the MADIS project (NOAA) [National Oceanic and Atmospheric Administration – NOAA 2008], and the Open Geospatial Consortium [Open Geospatial Consortium (OGC) 2008].

Plans for two additional case studies exist. The second case study deals with the data interoperability problem associated with human capital management classification information across heterogeneous consumers of human capital and types of industries (e.g., legal, administrative, medical, information technology, etc.)

The third case study aims at tackling some of the reported problems on the actual implementation of the Health Insurance Portability and Accountability Act–HIPAA [US Department of Health and Human Services 2008]. Recent recommendations note the need for software support to “exchange large quantities of data between diverse stakeholders, mapping, tracking, transforming, linking, and messaging.” [Charlene Underwood (on behalf of Siemens) 2008] Also “HIPAA regulations cannot supercede a contrary provision of State law if the State law imposes more stringent requirements. The resultant variation among state laws may impede interoperability, particularly when HIE [Health Information Exchange] crosses state lines.” [National Committee on Vital and Health Statistics 2008].

5. CONCLUSIONS

With the recent rise and success of multi-user immersive environments, such as the ones enabled by Second Life, the Web demonstrates its immense potential as a catalyst that promotes rich interactions and cooperations among global diverse communities. One might say there are some parallels between the evolution of these environments and that of the Web, as paradigms of communication and computation, namely: the level of development of the former today compares to that of the latter about 20 years ago when Mosaic was the preeminent browser. Unlike the Web, which is mainly about providing information to others—in the form of documents and artifacts—multi-user immersive environments are about facilitating interactions among users to support co-creation of knowledge, information, and associated artifacts. We argue that these environments provide an appropriate infrastructure for people to create knowledge collaboratively through activities such as the sharing, visualizing, and manipulating of data. In this paper we have described the architectural vision, its components, and related technologies of our initiative which aims at helping end-users to solve instances of the data interoperability problem by collaborating through a multi-user immersive environment and by capitalizing on resources these kinds of environments can automatically generate for them.

REFERENCES

- ALTOVA. 2008. MapForce. http://www.altova.com/products/mapforce/data_mapping.html.
- AMERICAN NATIONAL STANDARDS INSTITUTE. 2008. <http://www.ansi.org/>.
- AMIT, S. 2004. From Semantic Search and Integration to Analytics. In *Dagstuhl Seminar on Semantic Interoperability and Integration*, Y. Kalfoglou, M. Schorlemmer, A. P. Sheth, S. Staab, and M. Uschold, Eds. Schloss Dagstuhl.
- BIRSAN, D. 2005. On plug-ins and extensible architectures. *ACM Queue* 3, 2 (March).
- BRETHERTON, F. P. AND P. T. SINGLEY. 1994. Metadata: a user's view. In *Proceedings of International Working Conference on Scientific and Statistical Database Management*.
- CALVANESE, D., G. D. GIACOMO, M. LENZERINI, D. NARDI, AND R. ROSATI. 2001. Data Integration in Data Warehousing. *International Journal of Cooperative Information Systems (IJCIS)* 10, 3:237–271.
- CARNEY, D., D. FISHER, E. MORRIS, AND P. PLACE. 2005. Some Current Approaches to Interoperability. Tech. rep., Integration of Software-Intensive Systems Initiative, SEI, CMU.
- CHARLENE UNDERWOOD (ON BEHALF OF SIEMENS). 2008. NHIN Requirements: A Vendor Perspective. <http://www.ncvhs.hhs.gov/060727p14a.pdf>. A testimony to the National Committee on Vital and Health Statistics, Subcommittee on Standards and Security. July 2006.
- CHAUDHURI, S. AND U. DAYAL. 1997. An overview of data warehousing and OLAP technology. *ACM SIGMOD Record* 26, 1 (January):65–74.
- COMMON LOGIC WORKING GROUP WEBSITE. 2008. <http://cl.tamu.edu/>.
- DELL'ERBA, M., O. FODOR, F. RICCI, AND H. WERTHNER. 2002. Harmonise: A Solution for Data Interoperability. In *Proceedings of (Lisbon, Portugal)*. IFIP Conference on Towards The Knowledge Society: E-Commerce, E-Business, and E-Government.
- DOERR, M. 2004. The CIDOC CRM, an Ontological Approach to Schema Heterogeneity. In *Dagstuhl Seminar on Semantic Interoperability and Integration*, Y. Kalfoglou, M. Schorlemmer, A. P. Sheth, S. Staab, and M. Uschold, Eds. Schloss Dagstuhl.
- ECLIPSE PROJECT. 2008. Eclipse Integrated Development Environment. <http://www.eclipse.org/>.
- FACEBOOK DEVELOPERS. 2008. <http://developers.facebook.com/>.
- FLORIDA HURRICANE ALLIANCE. 2008. <http://www.ihrca.fiu.edu/fha/>.

- GRUBER, T. R. 1993. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition* 5, 2 (February):199–220.
- HAYES, P. 2008a. IKL – A Logic For Interoperation. http://ontolog.cim3.net/file/resource/presentation/PatHayes_20061026/OntologyWorkshopSlides.html.
- HAYES, P. 2008b. IKL Guide. <http://www.ihmc.us/users/phayes/IKL/GUIDE/GUIDE.html>.
- HOFMANN, M. A. 2004. Challenges of Model Interoperation in Military Simulations. *SIMULATION* 80, 12:659–667.
- IEEE COMPUTER SOCIETY. 2001. *Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*. IEEE Press, Los Alamitos Circle, California.
- IEEE-USA. 2005. Interoperability for the national health information network. http://www.ieeeusa.org/policy/positions/NHINinteroperability.html#_ftn1.
- INMON, W. H. 1992. *Building the Data Warehouse*. John Wiley & Sons, Inc.
- ISO/IEC. 2008. ISO/IEC 24707:2007 – Common Logic (CL): A Framework for a Family of Logic-Based Languages. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=39175.
- KAPTELININ, V. AND M. CZERWINSKI, Eds. 2007. *Beyond the Desktop Metaphor – Designing Integrated Digital Work Environments*. MIT Press, Cambridge, Massachusetts.
- KRIENS, P. 2008. How osgi changed my life. *ACM Queue* 6, 1 (January-February).
- LENZERINI, M. 2002. Data Integration: A Theoretical Perspective. In *Proceedings of ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*.
- LINDEN LAB. 2008a. Second Life. <http://secondlife.com/>.
- LINDEN LAB. 2008b. Second Life Land. <http://secondlife.com/land/index.php>.
- LINDEN LAB. 2008c. Second Life Resources for Developers. http://wiki.secondlife.com/wiki/LSL_Portal.
- MCGUINNESS, D. L., P. FOX, L. CINQUINI, P. WEST, J. GARCIA, J. L. BENEDICT, AND D. MIDDLETON. 2008. Enabling Scientific Research Using an Interdisciplinary Virtual Observatory: The Virtual Solar-Terrestrial Observatory Example. *AI Magazine* 29, 1 (Spring):65–76.
- METEORA AND OKEANOS ISLANDS. 2008. <http://slurl.com/secondlife/Meteora/177/161/27/>; <http://slurl.com/secondlife/Okeanos/64/217/30/>.
- MICROSOFT. 2008. SQL Server Integration Services – SSIS. <http://www.microsoft.com/sql/technologies/integration/default.aspx>.
- NASA'S SPACE CoLAB. 2008. <http://spacecolab.wordpress.com/>.
- NATIONAL COMMITTEE ON VITAL AND HEALTH STATISTICS. 2008. Enhanced Protections for Uses of Health Data. <http://www.ncvhs.hhs.gov/071221lt.pdf>. December 2007.
- NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION – NOAA. 2008. Meteorological Assimilation Data Ingest System (MADIS). <http://madis.noaa.gov/>.
- NETBEANS PLUGIN PORTAL. 2008. <http://plugins.netbeans.org/PluginPortal/faces/MainPage.jsp>.
- OPEN GEOSPATIAL CONSORTIUM (OGC). 2008. <http://www.opengeospatial.org/>.
- OPENSIMULATOR WIKI. 2008. http://opensimulator.org/wiki/Main_Page.
- OSGI ALLIANCE. 2008. <http://www.osgi.org/Main/HomePage>.
- PAEPCKE, A., C.-C. K. CHANG, T. WINOGRAD, AND H. GARCIA-MOLINA. 1998. Interoperability for digital libraries worldwide. *Communications of the ACM* 41, 4 (April):33–42.
- PAGE, E. H., R. BRIGGS, AND J. A. TUFAROLO. 2004. Toward a Family of Maturity Models for the Simulation Interconnection Problem. In *Spring Simulation Interoperability Workshop*. Arlington, VA.
- PAN, A. AND A. VIÑA. 2004. An alternative architecture for financial data integration. *Communications of the ACM* 47, 5 (May):37–40.
- PARENT, C. AND S. SPACCAPIETRA. 2000. Database Integration: The Key to Data Interoperability. In *Advances in Object-Oriented Data Modeling*, M. P. Papazoglou, Z. Tari, and S. Spaccapietra, Eds. MIT Press, 221–253.
- POWER, L. R. 1990. Post-Facto Integration Technology: New Discipline for and Old Practice. In *Proceedings of IEEE Conference on Systems Integration (ICSI'90)*. 4–13.

- RIEDEMANN, C. AND C. TIMM. 2003. Services for data integration. *Data Science Journal* 2, 26 (February):90–99.
- RIFAIEH, R. AND N. A. BENHARKAT. 2002. Query-based data warehousing tool. In *Proceedings of ACM International Workshop on Data Warehousing and OLAP*. ACM.
- SCI LANDS VIRTUAL CONTINENT. 2008. <http://www.sciland.org/>.
- SELIGMAN, L. AND A. ROSENTHAL. 2001. XML's impact on databases and data sharing. *Computer* 34, 6:59–67.
- SKIBO, C., M. YOUNG, AND B. JOHNSON. 2006. *Working With Microsoft Visual Studio 2005*. Microsoft Press.
- SMITH, M. K., C. WELTY, AND D. L. MCGUINNESS, Eds. 2004. *OWL Web Ontology Language Guide*. W3C.
- SOME MASHUP RESOURCES. 2008. <http://www.deitel.com/ResourceCenters/Web20/Mashups/tabid/977/Default.aspx>; <http://www.ibm.com/developerworks/library/x-mashups.html>; <http://blog.programmableweb.com/2007/05/07/12-useful-ruby-mashup-resources/>.
- STAHL, G. 2006. *Group Cognition – Computer Support for Building Collaborative Knowledge*. MIT Press, Cambridge, Massachusetts.
- STEP TOOLS INC. 2008. <http://www.steptools.com/>.
- SUN MICROSYSTEMS. 2008a. Project Darkstar. <http://projectdarkstar.com/>.
- SUN MICROSYSTEMS. 2008b. Project Wonderland. <https://lg3d-wonderland.dev.java.net/>.
- SÁNCHEZ-RUIZ, A. J. AND G. HANSEN. 2005. Translation patterns to specify processes in the PSL ontology. In *Proceedings of the sixth OOPSLA Workshop on Domain-Specific Modeling* (San Diego, CA). ACM-SIGPLAN.
- SÁNCHEZ-RUIZ, A. J., M. SAEKI, B. LANGLOIS, AND R. PALANO. 2007. Domain-specific software development terminology: Do we all speak the same language? In *Proceedings of the seventh OOPSLA Workshop on Domain-Specific Modeling* (Montreal, Canada). ACM-SIGPLAN.
- SÁNCHEZ-RUIZ, A. J. 1995. On automatic approaches to multi-language programming via code reusability. Ph.D. thesis, Rensselaer Polytechnic Institute. Available from <http://www.il.proquest.com>; UMI number: 9544074.
- TECHNET. 2001. Why Is Interoperability Important? [http://technet.microsoft.com/en-us/library/bb463148\(TechNet.10\).aspx](http://technet.microsoft.com/en-us/library/bb463148(TechNet.10).aspx).
- TOLK, A. AND J. A. MUGUIRA. 2003. The Levels of Conceptual Interoperability Model. In *Fall Simulation Interoperability Workshop*. Orlando, FL.
- TURNITSA, C. D. 2005. Extending the levels of conceptual interoperability model. In *Summer Computer Simulation Conference*. Philadelphia, PA.
- UMAPATHY, K. AND S. PURAO. 2006. Designing Enterprise Solutions with Web Services and Integration Patterns. In *Proceedings of IEEE International Conference on Services Computing (SCC)*. IEEE-CS.
- US DEPARTMENT OF DEFENSE. 2005. Defense standarization program. <http://www.dsp.dla.mil/>.
- US DEPARTMENT OF HEALTH AND HUMAN SERVICES. 2008. Health Insurance Portability and Accountability Act. <http://www.hhs.gov/ocr/hipaa/>.
- US DEPARTMENT OF TRANSPORTATION. 2008. National Surface Transportation Weather Observing and Forecasting System (Clarus). <http://www.its.dot.gov/clarus/index.htm>.
- VASSALOS, V. 2004. Interesting Problems in Semantic Integration and Interoperability. In *Dagstuhl Seminar on Semantic Interoperability and Integration*, Y. Kalfoglou, M. Schorlemmer, A. P. Sheth, S. Staab, and M. Uschold, Eds. Schloss Dagstuhl.
- VISSER, U., H. STUCKENSCHMIDT, H. WACHE, AND T. VOGELE. 2000. Enabling Technologies for Interoperability. In *Proceedings of International Symposium of Computer Science for Environmental Protection* (Bonn, Germany).
- WEST, J. 2003. The Role of Standards in the Creation and Use of Information Systems. In *Proceedings of MIS Quarterly Special Issue Workshop on Standard Making: A Critical Research Frontier for Information Systems*.
- WILKES, J. 2003. The Integrated Data Environment: a new tool for interoperability and effective

data integration for command and control. *Journal of Telecommunications and Information Technology* 4.

WINTERS, L. S., M. M. GORMAN, AND A. TOLK. 2006. Next Generation Data Interoperability: It's all About the Metadata. In *Proceedings of Fall Simulation Interoperability Workshop* (Orlando, Florida).

ZHU, F., M. TURNER, I. KOTSIPOULOS, K. BENNETT, M. RUSSELL, AND M. BUDGEN. 2004. Dynamic data integration using Web services. In *Proceedings of International Conference on Web Services*.



Arturo Sánchez-Ruiz an Associate Professor with the School of Computing at UNF, has presented numerous papers at renowned international conferences on Computing, organized by ACM, IEEE-CS, IFIP, and Dagstuhl. Has participated in research and consulting projects for the National Institute of Standards and Technology (NIST), the Venezuelan Petroleum Research Center (INTEVEP), and companies which belong to the main Venezuelan Petroleum Company (PDVSA). Has served on various conference program committees, National Science Foundation review panels, and has been a reviewer for the IEEE Latin America Transactions (since its inception), IEEE Software, and the Journal of Systems and Software. Dr. Sánchez-Ruiz earned a PhD degree in Computer Science from Rensselaer Polytechnic Institute and is a member of ACM, IEEE-CS, and AAAI.



Karthikeyan Umapathy is currently working as Assistant Professor of Information Systems at the School of Computing, University of North Florida.

He received his PhD in Information Sciences and Technology from the Pennsylvania State University. His research interests are interoperability among information systems, service-oriented computing, web services, systems integration, and IT standardization. His research works are published in various conferences such as Americas Conference on Information Systems, Conceptual Modeling, Hawaii International Conference on System Sciences, International Conference on Web Services, Service Computing Conference, and Design Science Research in Information Systems and Technology; and journals such as European Journal of Information Systems, Information Systems Frontiers, and Journal of Computing and Information Science in Engineering.



Pat Hayes is well-known for his work in AI and, more recently, the Semantic Web. In early work he made basic advances in formal knowledge representation, pioneering what is now called 'ontology design' and the logical foundations of common-sense reasoning. More recently he was a member of the W3C working groups developing RDF, OWL and SPARQL, and authored the ISO Common Logic standard. At various times, Hayes has been secretary of AISB, chairman and trustee of IJCAI, associate editor of the Journal of Artificial Intelligence, a governor of the Cognitive Science Society and president of AAAI. He is a charter Fellow of AAAI and the Cognitive Science Society.